# CS140 Operating Systems and Systems Programming
## Midterm Exam

**February 9, 2007**

**(Total time = 50 minutes, Total Points = 50)**

**Name: (please print)______________________________**

**In recognition of and in the spirit of the Stanford University Honor Code, I certify that I will neither give nor receive unpermitted aid on this exam.**

**Signature:_______________________________________**

**This examination is close notes and close book. You may not collaborate in any manner on this exam. You have 50 minutes to complete the exam. Before starting, please check to make sure that you have all 8 pages.**

| | |
|---|---|
| **1** | |
| **2** | |
| **3** | |
| **4** | |
| **5** | |
| **6** | |
| **7** | |
| **Total** | |

Name:_______________

1. (8 points) Assume you are building a special operating system that requires executing the expression:

   z = F3(F1(x), F2(F3(y)));

   where x, y, and z are integers and F1, F2, F3 are functions.

   The functions F1() and F3() must execute on thread T1 while the function F2 needs to execute on thread T2. Your job is to write the code to force the expression to be evaluated regardless of the order they are run by the CPU scheduler. Note you will have to add code to all three functions below.

   **Shared Variables:**

   ```
   int z, x, y;
   ```

   //Declare shared variables and semaphore with initial values here.

```
void ComputeZ() {



  StartThread(T1);



  StartThread(T2);



  return;
}
```

```
void T1() {

}
```

```
void T2() {

}
```

2. (6 points) Does your solution to Problem 1 handle the following cases? If so, explain why. If not, explain the problem and how you could fix it.

   (a) Would your solution handle the case when ComputeZ() is called twice in a row like:

```
    x = …;  y = …;   // Set x and y arguments
    ComputeZ();
    printf("z = %d\n", z);
    x = …;  y = …;   // Set x and y arguments
    ComputeZ();
    printf("z = %d\n", z);
```

   (b) Would your solution handle the case when ComputeZ() is called twice from two different threads. For example:

```
 void Z1() {
    x = …;  y = …;   // Set x and y arguments
    ComputeZ();
    printf("z = %d\n", z);
}
void Z2() {
    x = …;  y = …;   // Set x and y arguments
    ComputeZ();
    printf("z = %d\n", z);
}
StartThread(Z1); StartThread)(Z2);
```

3. (8 points) Assume you have a system with a static priority CPU scheduler. Assume the scheduler supports preemption but not priority donation. Describe what the program below prints if the priorities are set such that T2 has a high priority, T1 has the middle priority, and T0 has the low priority. Assume the system starts with only T0 executing. Assume the semaphore `mutex` is initialized to 1.

```
void T0() {
 printf(“T0-S\n”);

 StartThread(T1);

 printf(“T0-E\n”);

}
```

```
void T1() {
 printf(“T1-S\n”);

 P(mutex);

 printf(“T1-1\n”);

 StartThread(T2);

 printf(“T1-2\n”);

 V(mutex);

 printf(“T1-E\n”);
}
```

```
void T2() {

 printf(“T2-S\n”);

 P(mutex);

 printf(T2-1\n”);

 V(mutex);

 printf(“T2-E\n”);

}
```

4. (6 points) What changes, if any, would happen to the order the printfs in problem 3 if priority donation were added to the CPU scheduler?

5. (8 points)
(a) In a memory management unit (MMU) that supports segmentation and paging, pages can be shared in two different ways. Describe the two ways.

(b) In a segmented system it is possible to map a segment into two processes such that both processes can read and write the segment while a third process could map the segment read-only. Is it possible to do the same mapping with a paged-based system? Justify your answer.

6. (7 points) Describe how an operating can compute references bits for page replacement on a machine that doesn’t support reference bits in hardware.

7. (7 points) Describe the problem with libraries that position independent code (PIC) solves.